

1/12

FIG. 1

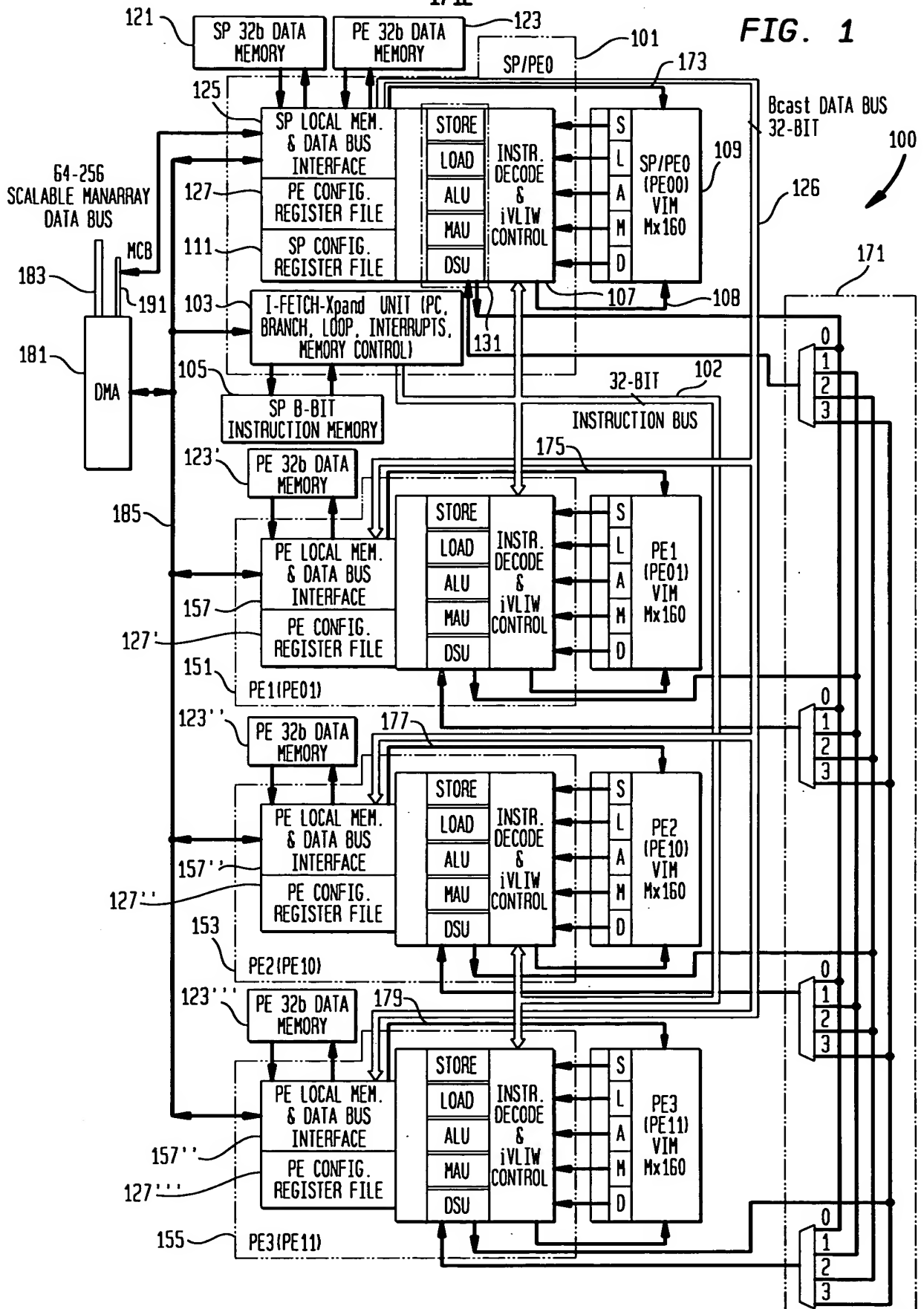


FIG. 2

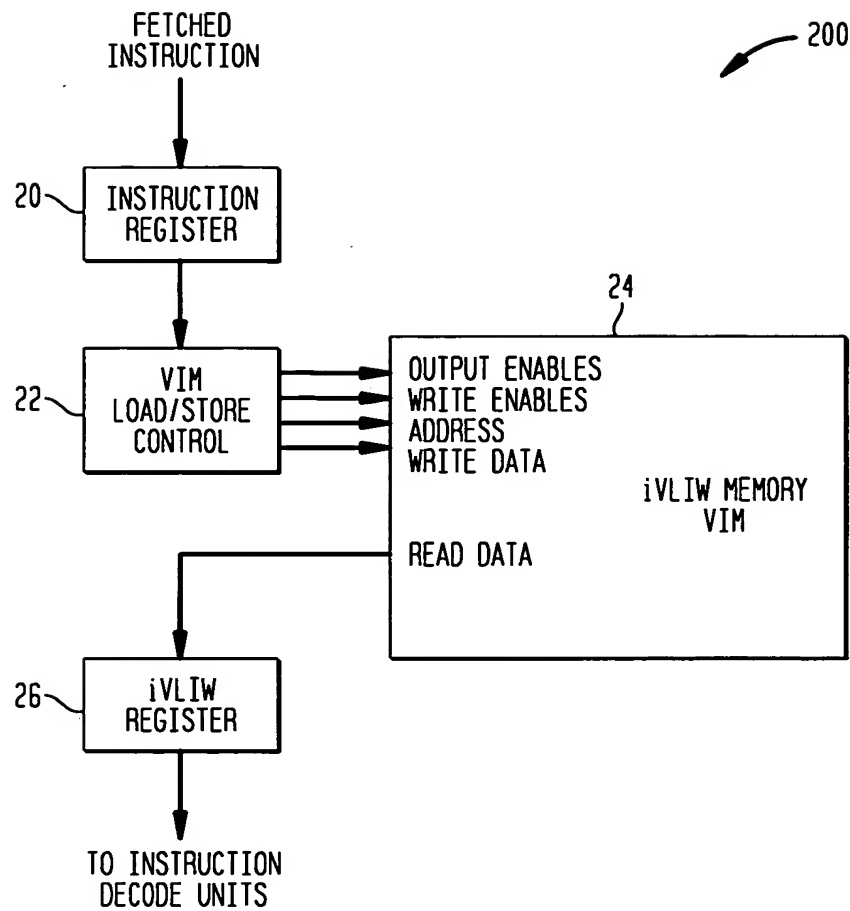


FIG. 3

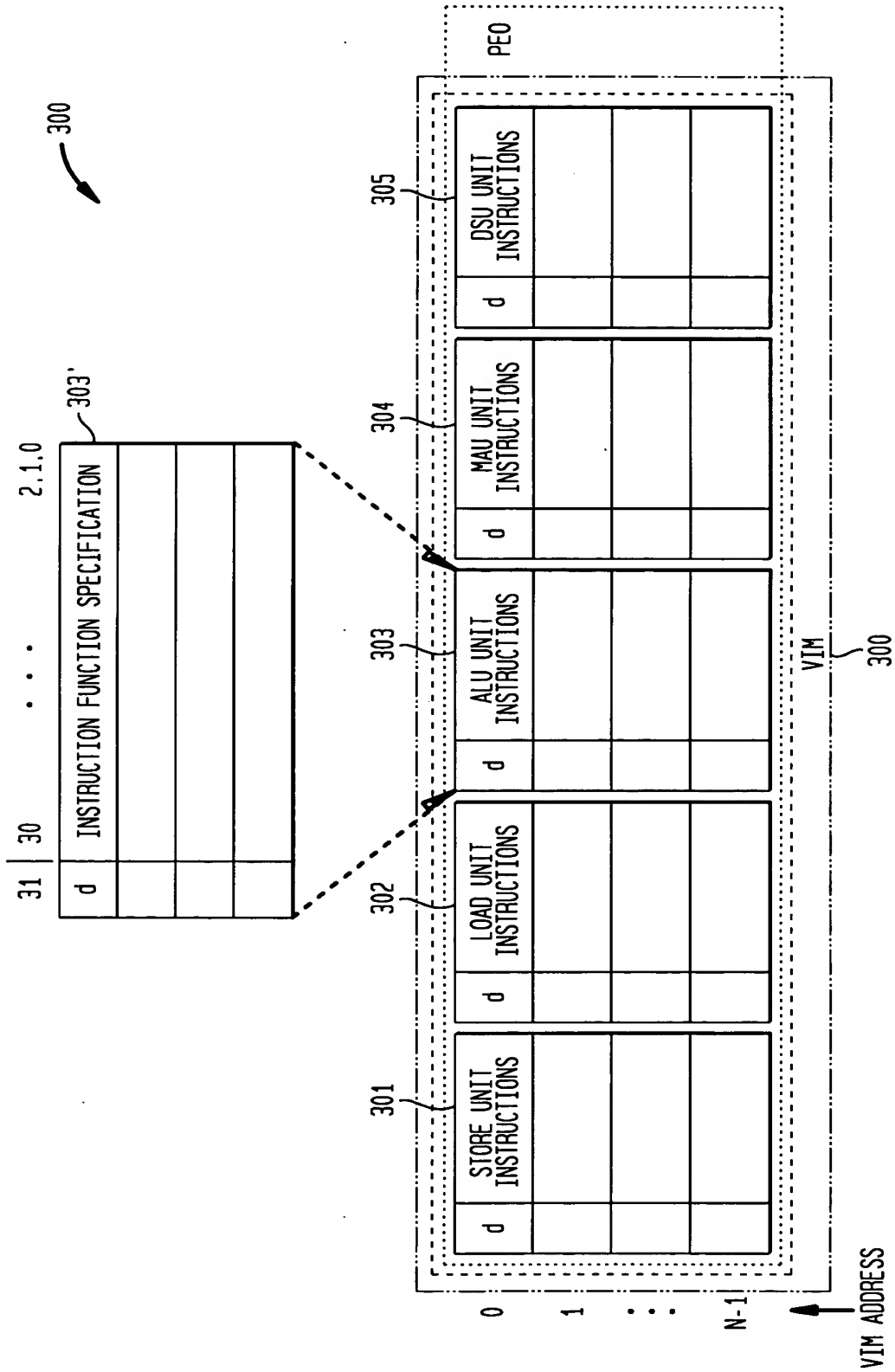


FIG. 4A

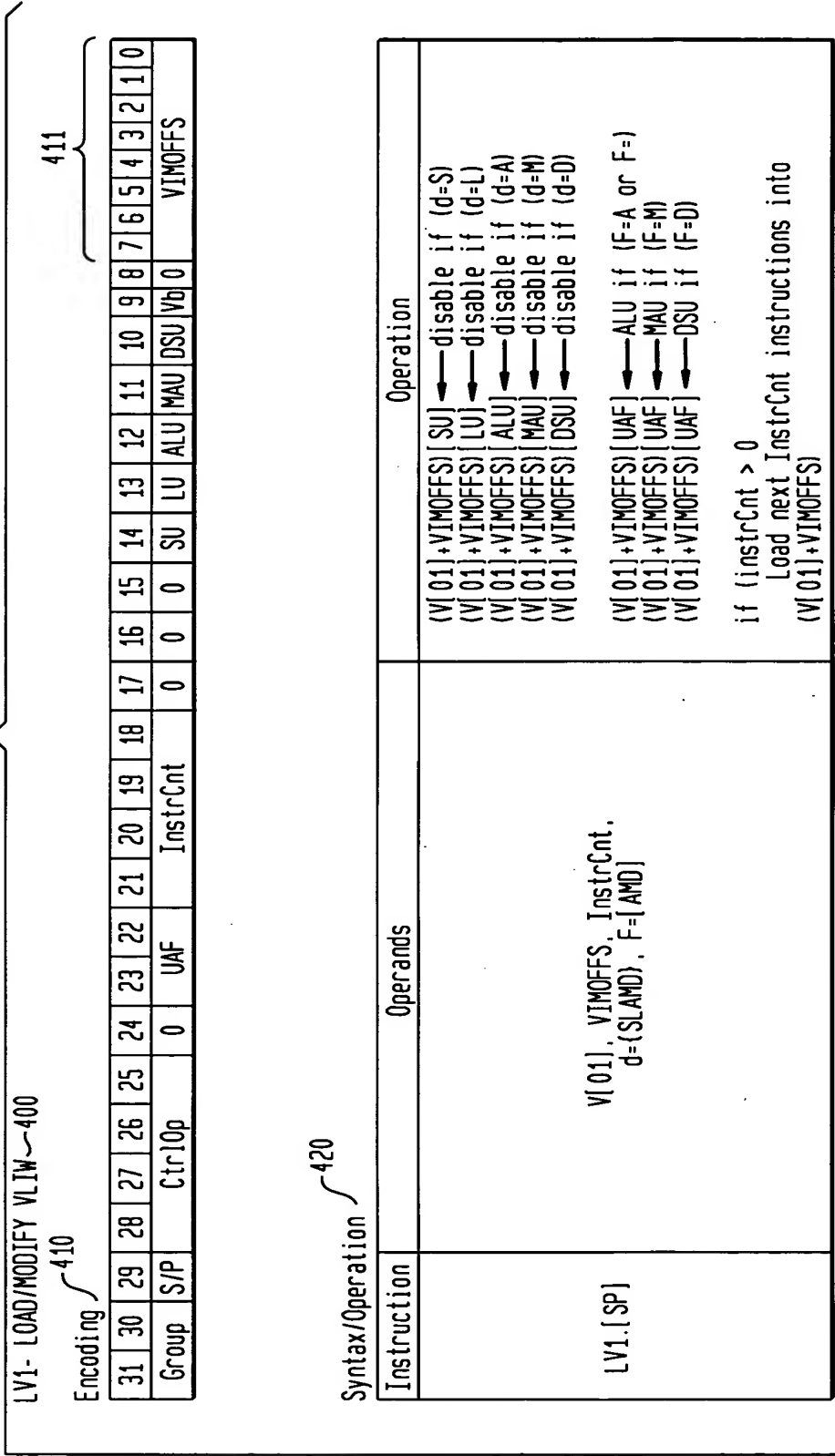
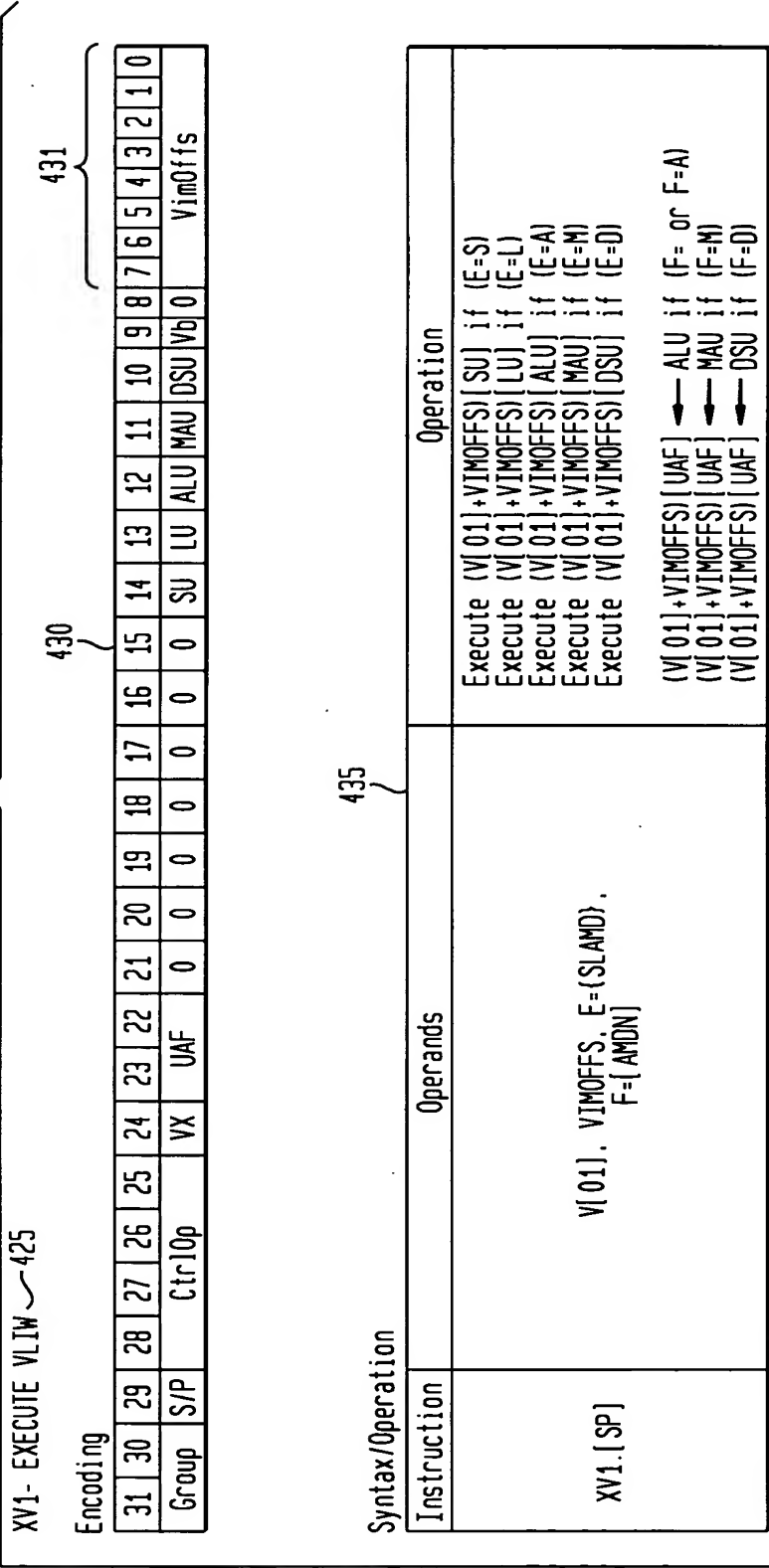


FIG. 4B



LV2- LOAD/MODIFY VLIW- 2~455

Encoding																																			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12			11	10	9	8	7	6	5	4	3	2	1	0		
Group	S/P	Ctrl0p				0		LI	d	InstrCnt				UnitVIM				0	0	0	0	0	0	0	Vb0	Vim0ffs									

Instruction	Operands	Operation
LV2.{SP}	LI, u=UnitVIM, V[01], VIMOFFS, InstrCnt, d	if (LI=0) Load disable bit only disable bit $\leftarrow (V[01] + VIMOFFS) [UnitVIM] \leftarrow d$ if (LI=1) Load instructions disable bit $\leftarrow (V[01] + VIMOFFS) [UnitVIM] \leftarrow d$ Load next InstrCnt instructions into $(V[01] + VIMOFFS) [UnitVIM] \leftarrow$ 1st Instruction following LV2 $(V[01] + VIMOFFS + 1) [UnitVIM] \leftarrow$ 2nd Instruction following LV2 .. $(V[01] + VIMOFFS + InstrCnt) [UnitVIM] \leftarrow$ (InstrCnt) th Instruction following LV2 InstrCnt is a binary coded number, 0 thru F, that represents from 1 to 16 instructions that can be loaded into up to 16 consecutive UnitVIM locations

XV2- EXECUTE VLIW ~ 475

Encoding			110																												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Group	S/P				Ctr10p			UAF	0	0	Vb			S0FS				L0FS									M0FS				D0FS

Syntax/Operation

Instruction	Operands	Operation
XY2.[SP]	V[01], SOFFS, LOFS, AOFFS, MOFS, DOFS, F={AMDN}	Execute (V[01]+SOFFS)[SU] if (d=0) Execute (V[01]+LOFFS)[LU] if (d=0) Execute (V[01]+AOFFS)[ALU] if (d=0) Execute (V[01]+MOFFS)[MAU] if (d=0) Execute (V[01]+DOFFS)[DSU] if (d=0)
		Execute NOP @ (V[01]+unit0FS)[unit] if (d=1)

8/12

FIG. 5

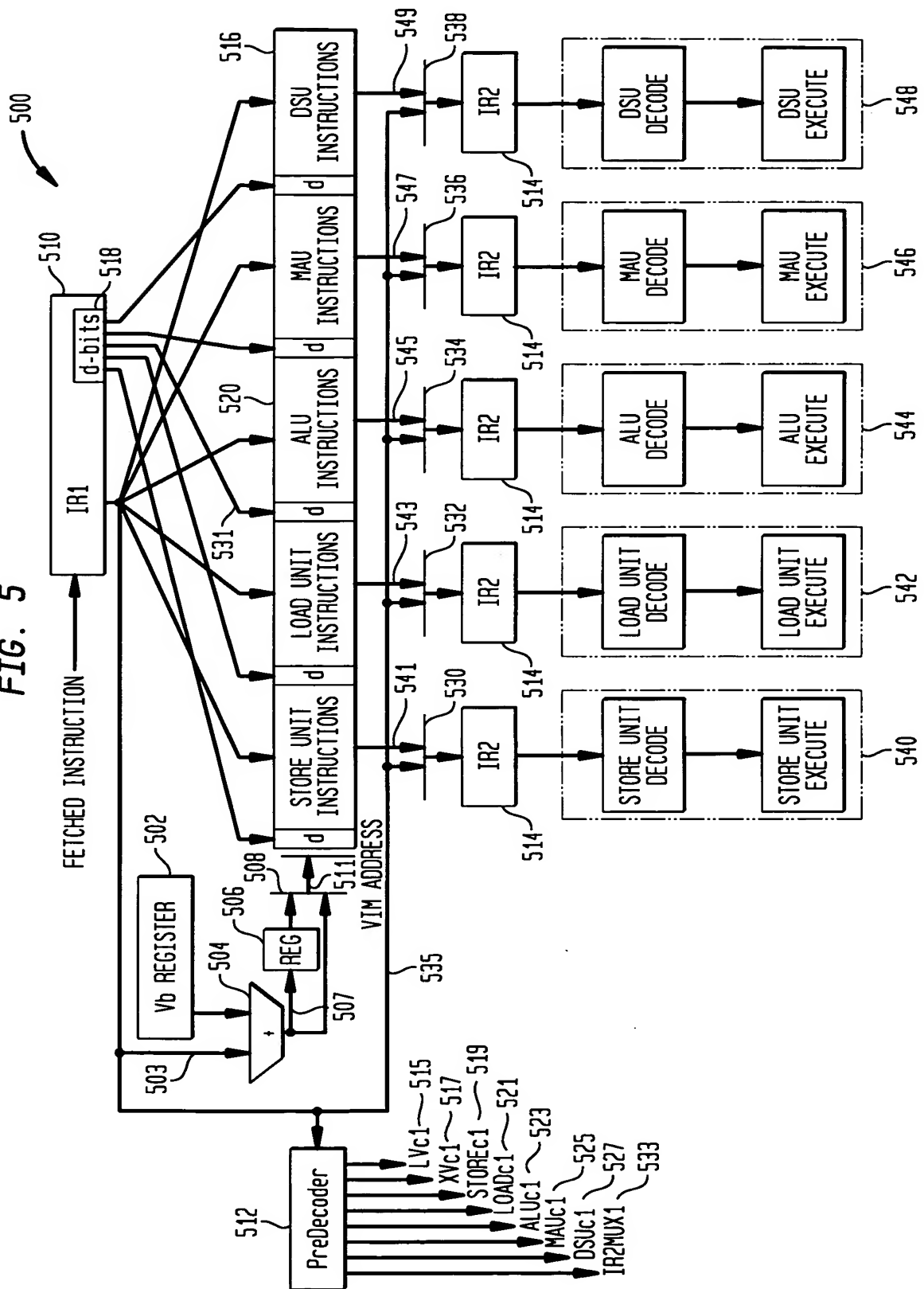


FIG. 6

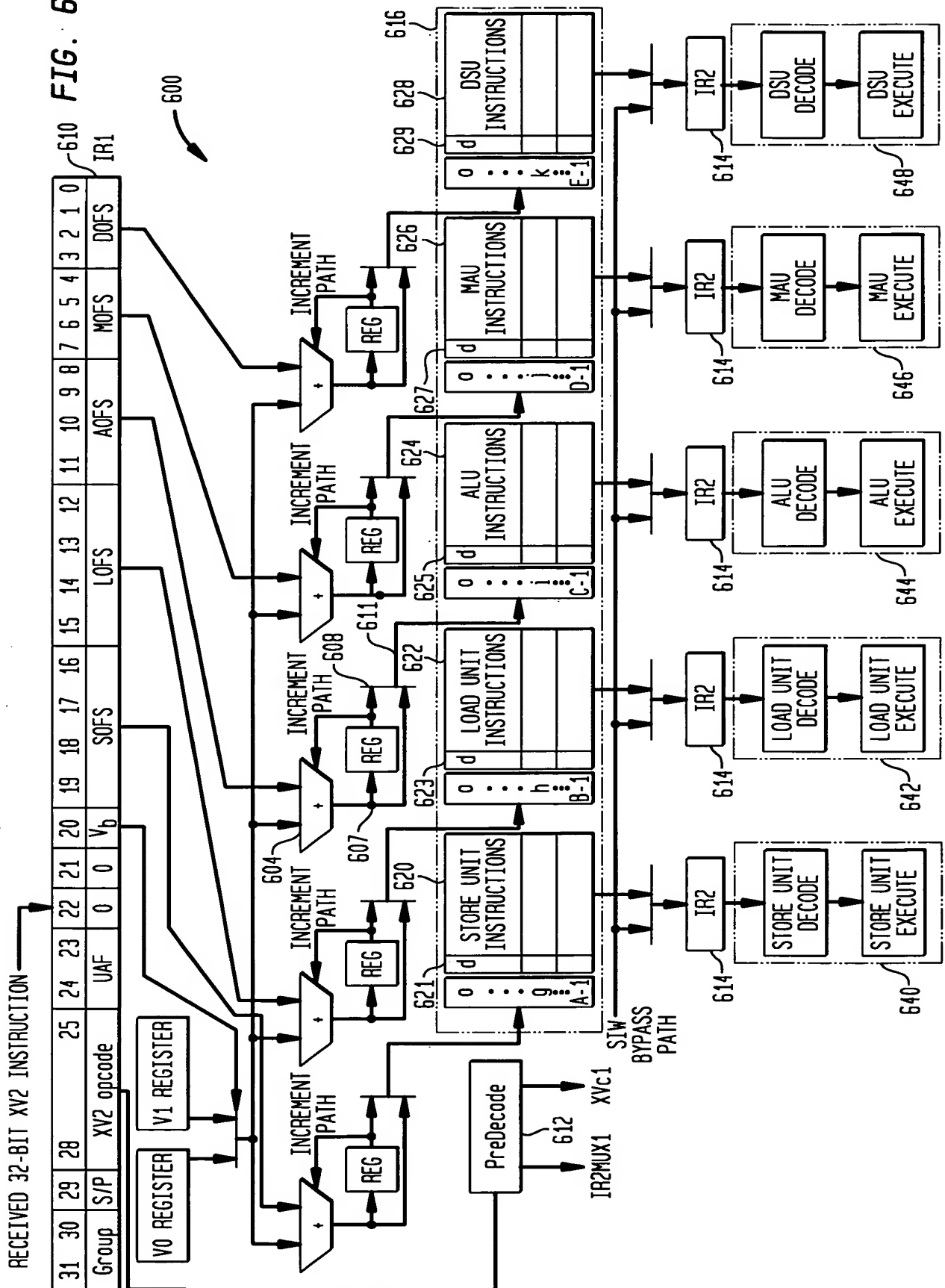


FIG. 7

